BLG202E Term Project - Image Compression via Truncated Singular Value Decomposition

Ömer Faruk San - 150220307
Department of AI and Data Engineering
Istanbul Technical University
Istanbul, Turkey
Email: san22@itu.edu.tr

***Abstract*—This report investigates the application of Truncated Singular Value Decomposition (TSVD) for image compression. Two images—a classical artwork and a natural photograph—were used to evaluate the efficiency of TSVD at varying approximation ranks. The implementation was done from scratch without using built-in SVD libraries. Experimental results demonstrate the trade-off between compression ratio and reconstruction error, with the Frobenius norm used to measure distortion.**

***Index Terms*—TSVD, image compression, SVD, Frobenius norm, compression**

I. Introduction

In the age of massive digital data, efficient image compression methods are vital for reducing storage and transmission costs. Singular Value Decomposition (SVD) is a powerful matrix factorization technique that can be adapted for image compression through its truncated form. By preserving only the most significant singular values, Truncated SVD (TSVD) provides a reduced-rank approximation of an image, balancing quality and compression.

II. Methodology

Given an image matrix $A \in \mathbb{R}^{m\times n}$, the full SVD is expressed as:

$$A = U\Sigma V^T \quad (1)$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{m\times n}$ is a diagonal matrix of singular values.

The truncated version retains only the top r singular values:

$$A_r = U_r\Sigma_r V_r^T \quad (2)$$

with $U_r \in \mathbb{R}^{m\times r}$, $\Sigma_r \in \mathbb{R}^{r\times r}$, and $V_r \in \mathbb{R}^{n\times r}$.

For RGB images, TSVD was applied independently on each color channel. The reconstructed image was then obtained by merging the approximated R, G, and B matrices.

III. Implementation

The TSVD algorithm was implemented entirely from scratch in Python without relying on high-level decomposition libraries such as `numpy.linalg.svd` or `scipy.linalg.svd`, in line with the assignment constraints.

To approximate the SVD of each channel, the algorithm begins by forming the covariance matrix A^TA, and then iteratively estimates the dominant eigenvectors and eigenvalues using the power iteration method. Once the top eigenvector is found, a deflation step is applied to remove its influence, allowing the next eigenvector to be calculated. This process continues until r eigenvectors are obtained.

This procedure is repeated for each of the R, G, and B channels of the input image, which are first extracted and converted into two-dimensional grayscale-like matrices. Each channel is decomposed and reconstructed individually. The reconstructed R, G, and B channels are then merged back into a final RGB image.

IV. Experimental Results

A. Input Images

Two images were used, shown in Figure 1:

(a) Brueghel painting

(b) Mandrill face

Fig. 1: Original input images used for TSVD compression

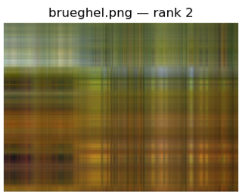
- **Brueghel painting**: a detailed classical landscape with complex textures.
- **Mandrill face**: a high-contrast, vivid photo with strong edges.

B. Visual Results

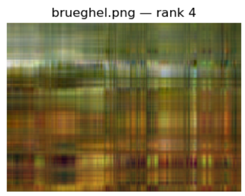
We present the reconstructed images across various ranks for both test images. This provides a visual comparison of how image quality evolves as more singular values are retained during TSVD. Each image was reconstructed at ranks $r = 2, 4, 8, 16, 32, 64$. The goal is to observe how well the essential features of the original image are preserved and how quickly useful visual information is recovered.

As shown in the figures, lower-rank approximations (e.g., $r = 2$ or $r = 4$) retain only coarse structures and colors, while higher ranks (e.g., $r = 32$ and $r = 64$) result in sharper and more detailed images. This reflects how TSVD prioritizes the most significant components of the image matrix and progressively restores detail as rank increases. Such progressive visual recovery is particularly noticeable in the Mandrill image, where edge features and high contrast patterns are quickly captured.

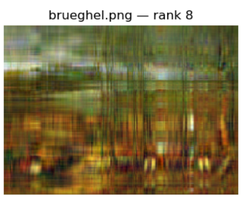
These visual results support the quantitative trends seen in the error plots and demonstrate that TSVD can achieve a balance between compression and quality, especially for images with structured content. r.



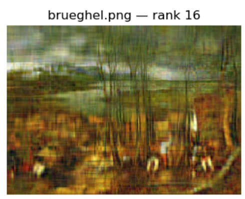
(a) $r = 2$



(b) $r = 4$



(c) $r = 8$



(d) $r = 16$

(e) $r = 32$

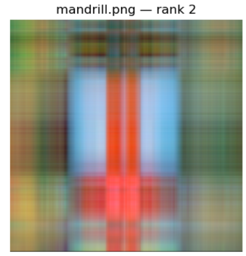
(f) $r = 64$

Fig. 2: Brueghel painting reconstructions at ranks 2, 4, 8, 16, 32, and 64.

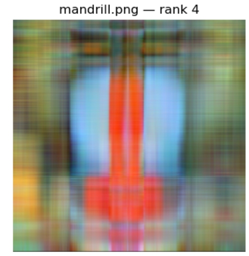
C. Error Analysis

To assess the quality of the compressed images, I used the Frobenius norm of the difference between the original image matrix A and the reconstructed matrix A_r:

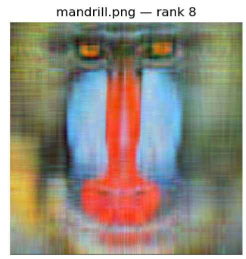
$$\|A - A_r\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(A_{ij} - (A_r)_{ij})^2} \tag{3}$$



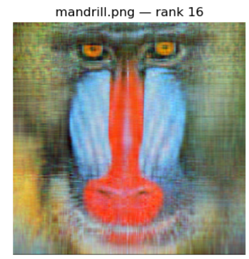
(a) $r = 2$



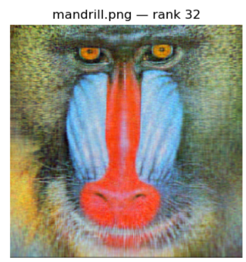
(b) $r = 4$



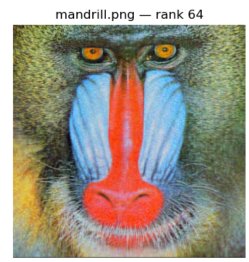
(c) $r = 8$



(d) $r = 16$



(e) $r = 32$



(f) $r = 64$

Fig. 3: Mandrill face reconstructions at ranks 2, 4, 8, 16, 32, and 64.

This norm provides a global scalar value representing the overall reconstruction error.

Each color channel (R, G, B) was processed separately, and their individual errors were computed. The final error reported for an image is the sum of the Frobenius norms across all three channels:

$$\text{Total Error} = \|A^R - A_r^R\|_F + \|A^G - A_r^G\|_F + \|A^B - A_r^B\|_F \tag{4}$$

Figure 4 plots this total error against various rank values for both images. As expected, the reconstruction error decreases monotonically with increasing r. The Mandrill image exhibits a faster error reduction compared to the Brueghel image, indicating that it can be more effectively compressed at lower ranks.

Additionally, I include individual error plots for each image to visualize their distinct behaviors:

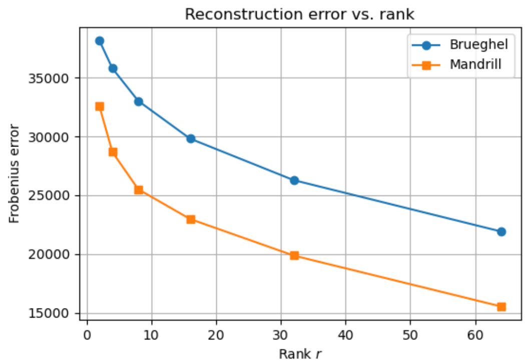
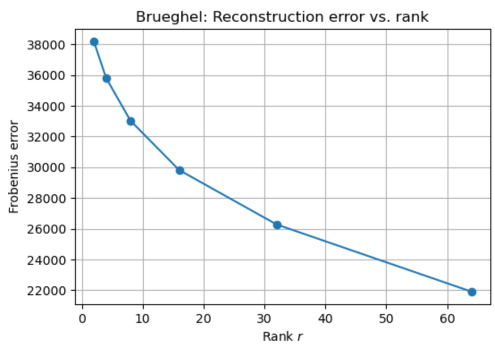
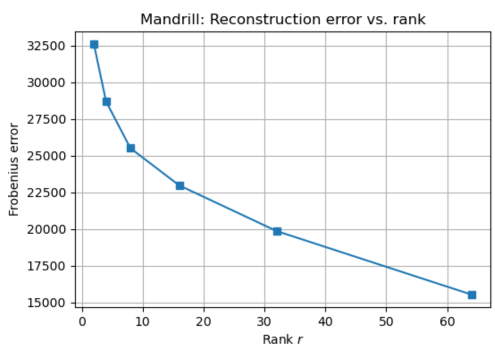


Fig. 4: Reconstruction error vs. rank for both images



(a) Brueghel Image



(b) Mandrill Image

Fig. 5: Per-image reconstruction error vs. rank

D. Storage Analysis

Let an image have size $m \times n$. Full RGB image storage requires $3mn$ values, accounting for the three color channels.

A TSVD-compressed version at rank r requires:

$$3r(m+n+1) \tag{5}$$

values in total. This includes r singular values and r columns of U and V matrices for each of the R, G, and B channels. Thus, TSVD allows linear control over storage cost by adjusting the rank r.

To illustrate the impact of rank on storage requirements, I plotted the number of elements required to store the original and TSVD-compressed images.

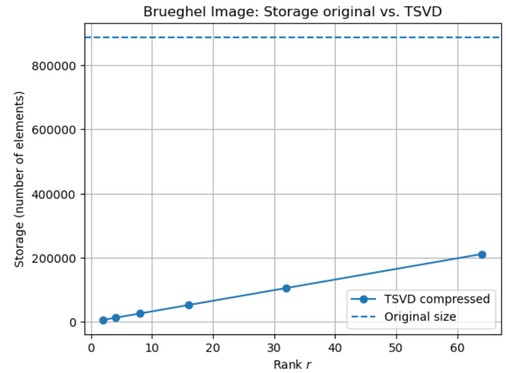


Fig. 6: Storage requirement comparison for the Brueghel image.

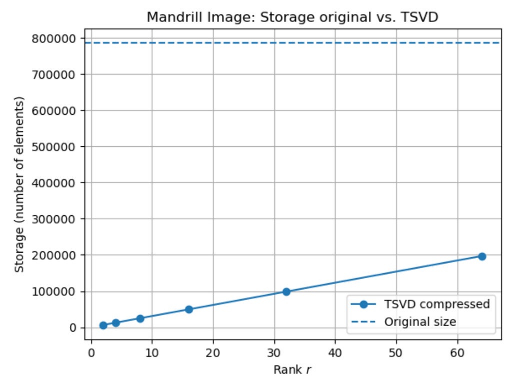


Fig. 7: Storage requirement comparison for the Mandrill image.

From these plots, it is evident that while the original images require a fixed high storage cost, TSVD-compressed images scale linearly with rank r. For instance, even at $r = 16$, the TSVD representation can save over 80%of storage while retaining acceptable visual quality. This trade-off is particularly valuable in environments with limited memory or bandwidth.

Moreover, this reduction in storage space comes with a corresponding increase in approximation error, as shown earlier. Therefore, the choice of rank should balance between visual accuracy and memory efficiency according to the application requirements.

V. Challenges

While TSVD proved to be a useful method for compressing structured images, implementing it from scratch without built-in decomposition tools posed several challenges. One of the main difficulties was constructing an SVD-like decomposition by relying only on iterative techniques such as power iteration and deflation. Ensuring numerical stability and accurate convergence across color channels required careful implementation and tuning.

Another challenge was managing image channel operations and reconstructing RGB images accurately after processing them independently. Since the performance and quality varied significantly between the two images, selecting appropriate ranks also involved subjective judgment based on visual and quantitative results.

Debugging the linear algebra components—especially when approximations were poor at low ranks—was particularly time-consuming. Lastly, the computational load increased noticeably with image size and rank, which required optimizing loops and avoiding redundant calculations.

VI. Discussion and Conclusion

From my experiments, I observed that the Mandrill image compressed much more effectively than the Brueghel painting, especially at low ranks. This makes sense when considering the structure of both images. The Mandrill image contains strong color blocks and sharp transitions, which are easier to capture using fewer singular values. On the other hand, the Brueghel painting is rich in detail, texture, and subtle color variations that require higher ranks to approximate accurately.

At rank $r = 16$, for instance, the Mandrill image was already quite visually clear, while the Brueghel image still appeared blurry and lacked fine structure. This was reflected in the Frobenius error plots: the Mandrill image had a steeper drop in error as the rank increased.

I also noticed that while increasing the rank always improved visual quality, the improvement rate decreased after a certain point. This is expected in compression tasks: the first few singular values capture the majority of the image's energy, and later components contribute more to detail than to overall structure.

Overall, the project showed me that TSVD can be a powerful compression technique for images with regular patterns or strong features, but it may struggle with more complex visuals unless a high rank is used. In such cases, the storage benefits decrease, so it becomes important to balance quality with compression needs.

References

[1] U. M. Ascher and C. Greif, *A First Course in Numerical Methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2011.

[2] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. Johns Hopkins University Press, 2013.